

# Transistor quickly wakes sleeping LDO

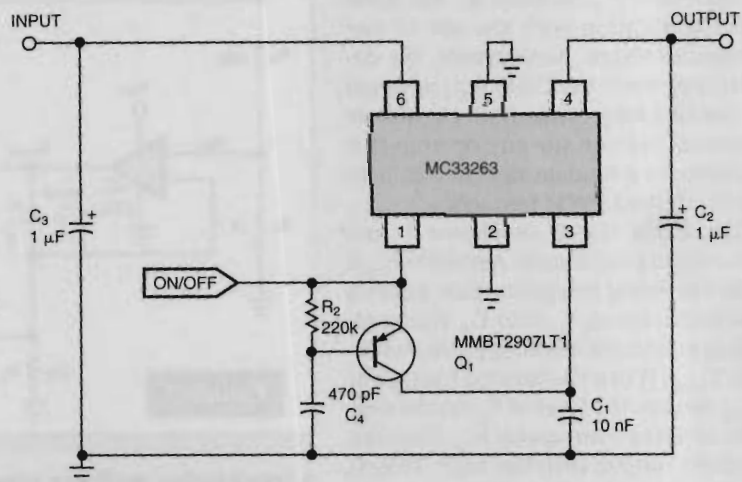
CHRISTOPHE BASSO, MOTOROLA SEMICONDUCTOR, TOULOUSE, FRANCE

Portable systems, such as telephone handsets, make extensive use of low-dropout (LDO) regulators. These components provide noise-sensitive parts with a stable power-supply line. When a telephone enters standby mode, most of the circuits go to sleep by disabling the LDO's outputs. Operating current thus drops to a minimal level. When a user starts to dial a number, the LDO receives an enable signal and immediately delivers the nominal operating voltage. Unfortunately, most low-noise LDOs use a bypass capacitor that briefly loads the internal reference voltage upon wake-up. In fact, the output exhibits a latency period before reaching its steady-state level. With a 10-nF bypass capacitor, this period typically lasts 1 msec and correspondingly degrades the overall response time. The ultra-low-noise MC33263 from Motorola ([www.motorola.com](http://www.motorola.com)) also uses a 10-nF bypass capacitor. However, the EZCap architecture of the IC allows the use of an inexpensive decoupling capacitor (ESR from 10 mΩ to 3Ω) and allows the designer to speed the wake-up time (Figure 1).

The base of a low-cost pnp transistor connects to an RC network. At power-up,  $C_4$  discharges. When the control logic sends its high-going wake-up signal, the transistor's base is momentarily tied to ground. The transistor turns on and immediately charges bypass capacitor  $C_1$  toward its nominal operating voltage. After a few microseconds, the pnp turns off and becomes transparent to the regulator. This circuit dramatically improves the response time of the regulator from 1 msec to 30 μsec (Figure 2). You need only adjust the RC time constant to avoid any bypass-capacitor overload during the wake-up transient. Such an overload would generate an unacceptable output overshoot. Because the transistor connects to the bypass pin, it does not degrade the noise performance of the LDO. (DI #2241)

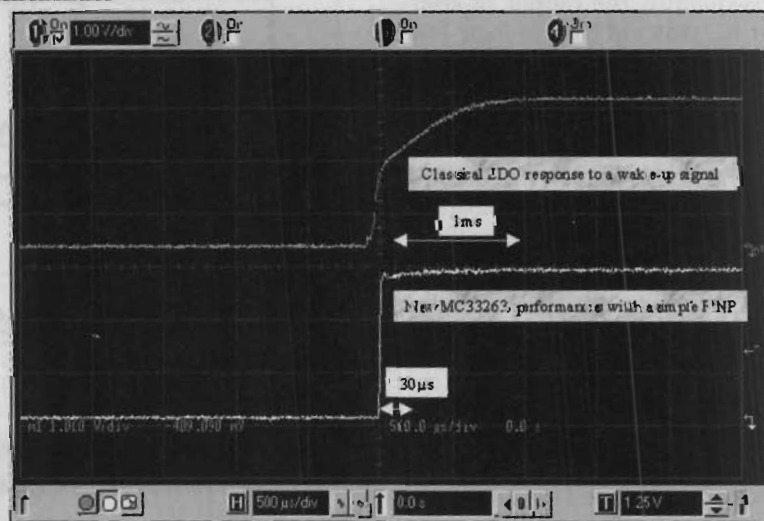
EDN

FIGURE 1



An inexpensive transistor and two passive components drastically improve the wake-up-response time of a linear regulator

FIGURE 2



To Vote For This Design, Circle No. 355

The simple circuit in Figure 1 improves the wake-up-response time of the LDO from 1 msec (upper trace) to 30 μsec (lower trace).

# Circuits provide 4- to 20-mA PWM control

TOM GAY, DARMSTADT, GERMANY

The circuits in Figures 1 and 3 are useful when you use 4- to 20-mA current-loop signals to control a PWM signal. In both circuits, the minimum pulse width (corresponding to a 4-mA loop current) and the maximum pulse width (corresponding to a 20-mA loop current) are independently adjustable in the dedicated application with the use of one reference voltage. Furthermore, the circuits shut down the PWM output signal in case of a loop break. Both circuits are low-cost; you can use any op amp that provides an adequate slew rate to handle the desired PWM frequency.

The PWM circuit in Figure 1 uses free-running oscillation. Amplifier IC<sub>1</sub> is a noninverting integrator that forces a constant current,  $I_{C_1}$ , into  $C_1$ , thus providing a constant linear slope on its output,  $V_{P_{PWM}}$ . When the divided fraction of  $V_{P_{PWM}}$  reaches the level of  $V_C$  on the negative input of comparator IC<sub>3</sub>, the comparator's output switches high. Hence, FET Q<sub>2</sub> short-circuits divider resistor R<sub>5</sub>, and the undivided level of  $V_{P_{PWM}}$  undergoes comparison with  $V_C$ . R<sub>7</sub> extends the turn-on time of Q<sub>1</sub> with respect to Q<sub>2</sub>, such that FET Q<sub>1</sub> discharges C<sub>1</sub> with minimum delay. The output of amplifier IC<sub>1</sub>,  $V_{P_{PWM}}$ , then rapidly slews down to the level of  $V_{R4}$ .

Consequently, the output of comparator IC<sub>3</sub> switches back to logic low, and IC<sub>1</sub> restarts charging C<sub>1</sub>. To ensure that IC<sub>3</sub> returns to a logic-low level, you must generally set  $V_C$  a few millivolts higher than  $V_{R4}$ . The divider network, R<sub>2</sub>, R<sub>3</sub>, and R<sub>4</sub>, guarantee this setting. You can use the output of comparator IC<sub>3</sub> as a trigger signal for synchronizing other circuits. Resistor R<sub>IN</sub> terminates the loop current, and comparator IC<sub>2</sub> provides the PWM circuit's output signal, PWM<sub>OUT</sub>, by comparing the current-loop signal  $V_{IN}$  with  $V_{P_{PWM}}$ . The circuit in Figure 1 has PWM<sub>OUT</sub> set to 0% duty cycle at 4-mA loop current and 80% at 20-mA loop current (Figure 2).

The circuit in Figure 3 is a simplified, gated version of the circuit in Figure 1, with a synchronization input. It works the same, with one difference: The output of amplifier IC<sub>1</sub>,  $V_{P_{PWM}}$ , rises to and remains at the positive supply rail for IC<sub>1</sub>. As before, FET Q<sub>1</sub> short-circuits C<sub>1</sub> with each rising edge on the Sync input,

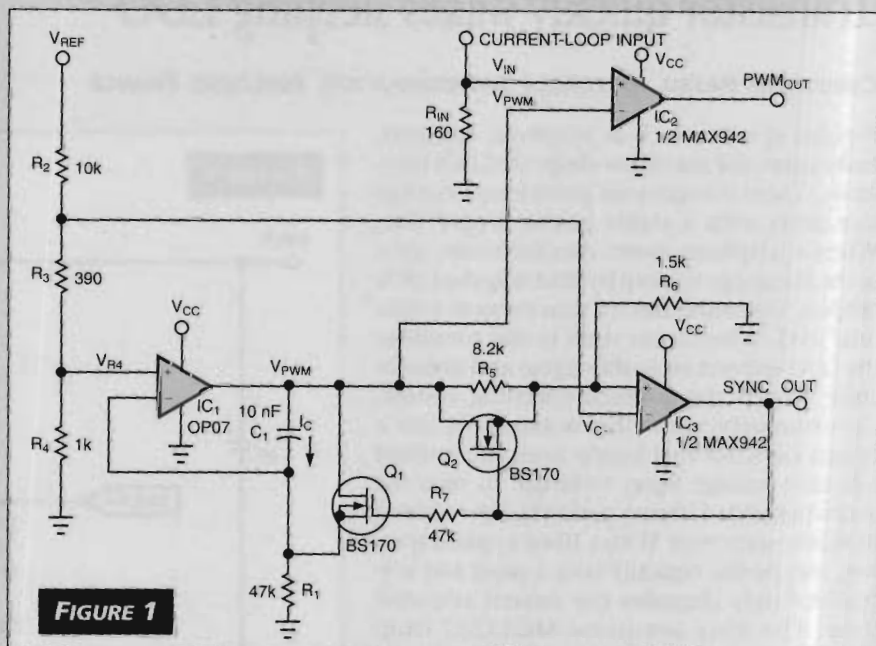


FIGURE 1

A free-running oscillator powers a current-loop controller for PWM signals. Duty cycle varies from 0 to 80% over the full-scale current range.

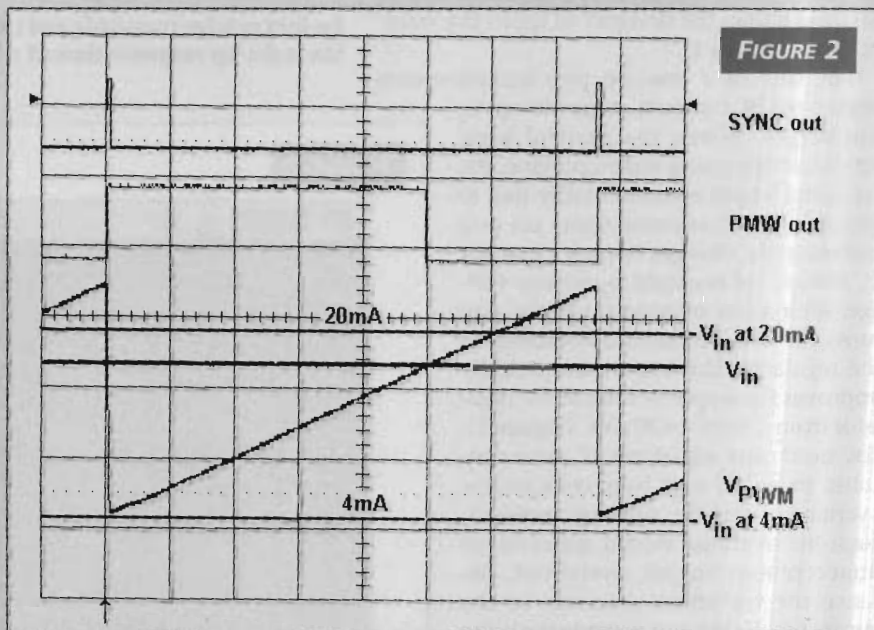


FIGURE 2

This simplified version of the circuit in Figure 1 has a synchronization input instead of an output.

causing IC<sub>1</sub> to slew  $V_{P_{PWM}}$  down to the level of  $V_{R4}$ . The network comprising R<sub>2</sub>, R<sub>3</sub>, and C<sub>2</sub> transforms a low-to-high transition on the Sync input into a narrow gate pulse, which turns on Q<sub>1</sub> for a short time. If you connect the Sync output of the

circuit in **Figure 1** to the Sync input of the circuit in **Figure 3**, you can omit  $R_2$ ,  $R_5$ , and  $C_2$ . The following formulas apply to **Figure 1**:

$$V_{R4} = V_{REF} \frac{R_4}{R_2 + R_3 + R_4}$$

$$V_{PWM(MIN)} = V_{R4} \frac{R_5 + R_6}{R_6}$$

$$I_C = \frac{V_{R4}}{R_1}$$

$$V_{C-} = V_{REF} \frac{R_3 + R_4}{R_2 + R_3 + R_4}$$

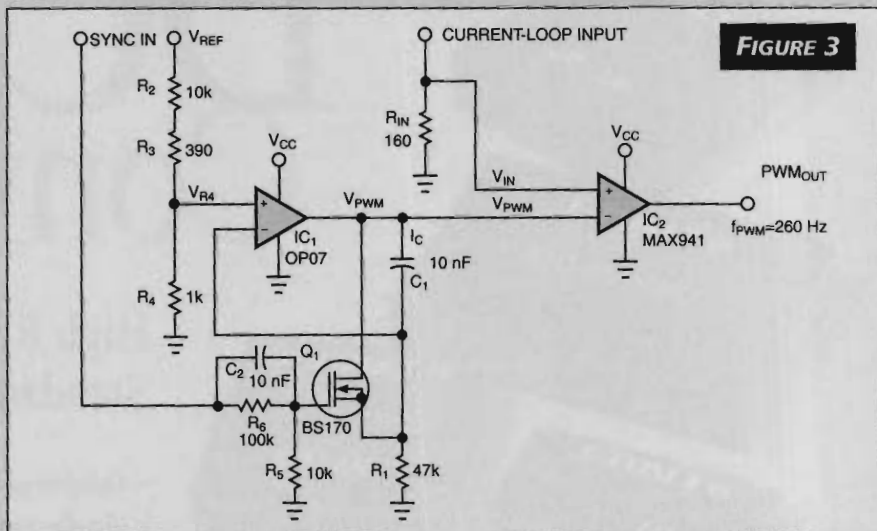
$$V_{PWM(MAX)} = V_{C-} \frac{R_5 + R_6}{R_6}$$

$$f_{PWM} = \frac{I_C}{C_1(V_{PWM(MAX)} - V_{PWM(MIN)})}$$

These equations apply to **Figure 3**:

$$V_{R4} = V_{REF} \frac{R_4}{R_2 + R_3 + R_4}$$

$$V_{PWM(MIN)} = V_{R4} \frac{R_5 + R_6}{R_6}$$



The circuit in **Figure 1** provides a linear duty-cycle-versus-current function.  $V_{PWM}$ , a function of the current, determines the duty cycle.

(DI #2242)

EDN

$$I_C = \frac{V_{R4}}{R_1}$$

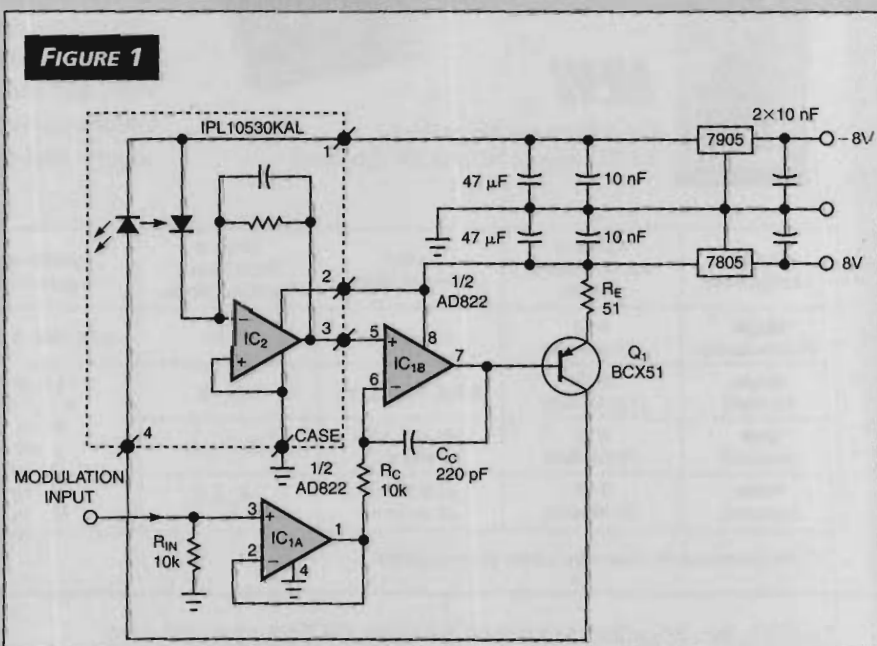
To Vote For This Design, Circle No. 356

## Circuit uses simple LED for near-IR light

LUKASZ ŚLIWCZYŃSKI, MARCIN LIPIŃSKI, INSTITUTE OF ELECTRONICS, KRAKOW, POLAND

You can successfully use LEDs as sources of near-infrared light. However, when you need a source of light with precisely controlled power, a feedback loop is necessary to compensate for the temporal and thermal changes of the LED parameters. Standard LED types come with neither these monitoring detectors nor an external monitoring photodiode to detect part of the emitted light and generate a feedback signal. The situation calls for some mechanical fixture to mount the photodiode. Such a solution, however, is bulky and cumbersome, especially when space is scarce. You can solve the problem by using an 880-nm IPL10530KAL hybrid detector/emitter module from Integrated Photomatrix Ltd. A modulated IR light source uses only one dual op amp, a transistor, and two voltage regulators (**Figure 1**).

The modulation input acts as a reference voltage and connects via amplifier IC<sub>1A</sub> to the comparing feedback-loop

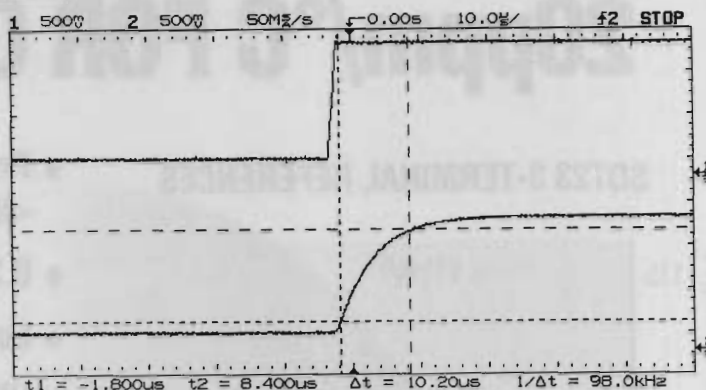


A few inexpensive components connected to a detector/emitter module enable you to obtain precise control of near-IR emitted light.

amplifier, IC<sub>1B</sub>. The resulting output voltage drives the output transistor, Q<sub>1</sub>, which directly drives the LED. Resistor R<sub>E</sub> limits the maximum drive current to approximately 80 mA, thus preventing damage to the LED. You can reduce R<sub>E</sub>'s value if you need higher power levels; the absolute maximum rating for the LED is 500 mA. If you change R<sub>E</sub>, you may need to alter the frequency-compensation network. The network comprises R<sub>C</sub> and C<sub>C</sub> and introduces a pole at 0 Hz and a zero at  $\frac{1}{2\pi R_C C_C}$  into the open-loop transfer function. The zero cancels the pole (at approximately 100 kHz) that the monitoring-photodiode preamplifier introduces, so the pole is a dominant pole in the feedback loop.

With the component values shown, the 3-dB modulation bandwidth of the source is approximately 40 kHz. You can experimentally determine the value of the compensating capacitor, C<sub>C</sub> by observing the voltage at Pin 3 of IC<sub>2</sub> and driving the circuit with a square wave (Figure 2). The output is a filtered version of the optical-output waveform. A Spice simulation shows the phase margin to be approximately 85°. The AD822 dual rail-to-rail op amp accommodates modulation-input voltage from 0 to 5V. The slope efficiency of the entire source (defined as  $dP_A/dV_{MOD}$ ) is approximately 1.5 mW/V and may vary slightly from unit to unit of detector/emitter modules.

FIGURE 2



IC<sub>2</sub> in Figure 1 delivers a clean response to a modulation-input step function.

You can use this design as an IR light source in a precise reflectometric measurement system incorporating pulse modulation and synchronous detection. To increase accuracy of the system comprising the entire optical head, you can install the system in a thermally stabilized environment with temperature controlled to within 0.5°C. The long-term measured power stability of the source is better than 1 ppm after initial warm-up. (DI #2243)

EDN

To Vote For This Design, Circle No. 357

## Two-DAC circuit adds and subtracts

V MANOHARAN, NAVAL PHYSICAL AND OCEANOGRAPHIC LABORATORY, KOCHI, INDIA

A typical way to add two binary words and provide an analog output is to use several digital ICs that drive a DAC. The circuit in Figure 1 eliminates the use of several digital IC packages and, hence, the need for the digital power supply. The circuit simultaneously carries out addition and subtraction on two 8-bit binary words and presents the output in bipolar analog form.

The hardware consists of four ICs, and the operation takes only 85 nsec, which is the settling time of the DACs plus the settling time of the op amp. IC<sub>1</sub>, a precision 10V reference, provides the reference current for both multiplying DACs, IC<sub>2</sub> and IC<sub>3</sub>. For these DACs,  $I_{REFA} = 10V/R_1$ , and  $I_{REFB} = 10V/R_2$ . In this case,  $I_{REFA} = I_{REFB} = I_{REF} = 2$  mA.

The output currents, I<sub>O<sub>A</sub></sub> and I<sub>O<sub>B</sub></sub>, depend on the respective A and B binary inputs and the input reference currents as follows:

$$I_{OA} = I_{REF} \cdot \left( \frac{N_A}{2^n} \right), \quad (1)$$

and

$$I_{OB} = I_{REF} \cdot \left( \frac{N_B}{2^n} \right), \quad (2)$$

where  $n$  is the number of input bits and  $N_A$  and  $N_B$  range in value from 0 to  $2^n - 1$ , in accordance with the input binary words.



The DAC-08 has complementary current outputs. Therefore, you can express the complements of  $I_{OA}$  and  $I_{OB}$  as

$$\overline{I_{OA}} = I_{FS} - I_{OA}, \quad (3)$$

and

$$\overline{I_{OB}} = I_{FS} - I_{OB}, \quad (4)$$

where  $I_{FS}$ , the full-scale current of the DAC, is

$$I_{FS} = \frac{2^n - 1}{2^n} \cdot I_{REF}. \quad (5)$$

The circuit configures IC<sub>4B</sub> as a current-to-voltage converter. Thus,

$$V_{OUT(A+B)} = I_{OA} \cdot R_5 + I_{OB} \cdot R_5. \quad (6)$$

Substituting  $I_{OA}$  and  $I_{OB}$  from Equations 1 and 2 into Equation 6 yields

$$\begin{aligned} V_{\text{OUT(A+B)}} &= I_{\text{REF}} \cdot \left( \frac{N_A}{2^n} \right) \cdot R_5 + I_{\text{REF}} \cdot \left( \frac{N_B}{2^n} \right) \cdot R_5 \\ &= \frac{I_{\text{REF}} \cdot R_5}{2^n} \cdot (N_A + N_B). \end{aligned} \quad (7)$$

$IC_{4A}$  serves as both a current-to-voltage converter for  $I_{OB}$  and a buffer to the potential drop across  $R_3$  because of the flow of  $I_{OA}$ . Thus, assuming  $R_3=R_4$ ,

$$\begin{aligned} V_{\text{OUT(A-B)}} &= \overline{I_{\text{OB}}} \cdot R_4 - \overline{I_{\text{OA}}} \cdot R_3 \\ &= (\overline{I_{\text{OB}}} - \overline{I_{\text{OA}}}) \cdot R_4. \end{aligned} \quad (8)$$

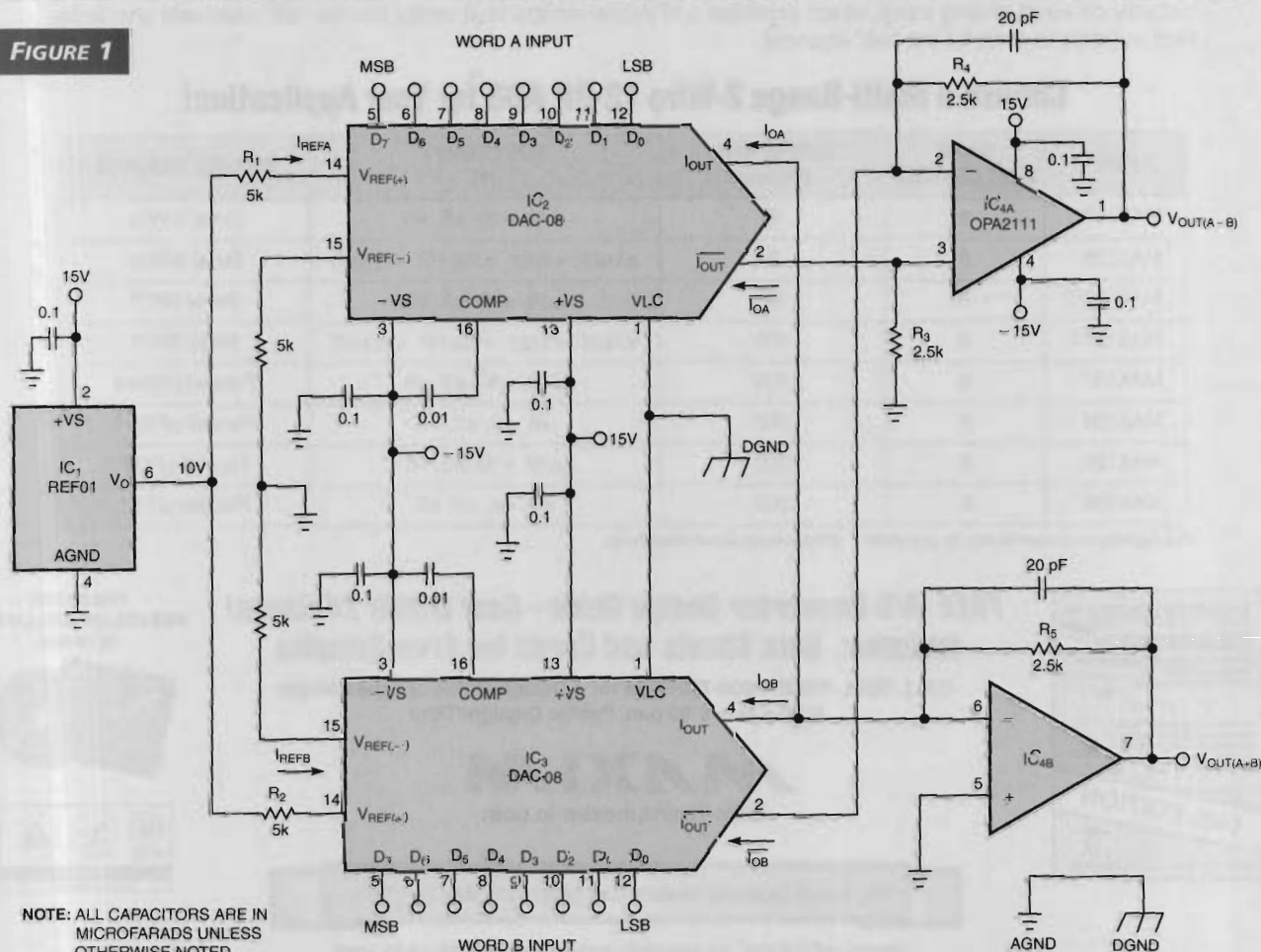
Substituting  $I_{OA}$  and  $I_{OE}$  from Equations 3 and 4 into Equation 8 and assuming that  $R_a = R_s$  result in

$$\begin{aligned} V_{\text{OUT(A-B)}} &= [(I_{\text{FS}} - I_{\text{OB}}) - (I_{\text{FS}} - I_{\text{OA}})] \cdot R_4 \\ &= (I_{\text{OA}} - I_{\text{OB}}) \cdot R_5 \\ &= \left[ I_{\text{REF}} \cdot \left( \frac{N_A}{2^n} \right) - I_{\text{REF}} \cdot \left( \frac{N_B}{2^n} \right) \right] \cdot R_5 \\ &= \frac{I_{\text{REF}} \cdot R_5}{2^n} \cdot (N_A - N_B). \end{aligned} \quad (9)$$

(DI #2226)

EDN

**To Vote For This Design, Circle No. 358**

**FIGURE 1**

**Two DACs and two op amps simultaneously carry out both addition and subtraction on two 8-bit binary words.**

# Calibration technique uses sound

AUBREY KAGAN, WEIDMULLER LTD, MARKHAM, ON, CANADA

A number of Weidmuller Ltd's ([www.weidmuller.com](http://www.weidmuller.com)) products require output calibration for zero and span. Normally, a technician calibrates a product by monitoring a digital readout and adjusting the zero and span potentiometers in turn. Because the technician must concentrate so heavily on the readout, the technician often experiences eyestrain, and the screwdriver often slips from the potentiometer. You can use an ADC card to implement the calibration function on a PC, and you can add an audio feature to simplify the calibration. In some cases, the use of audio can replace the need for a digital readout.

The audio spectrum arbitrarily divides into two sections: The first section ranges from 100 to 500 Hz; the second is 1 kHz and higher. The method compares the measured value from the device under test (DUT) with the desired value. If the measured value is lower than the desired value, the output is a frequency from the lower range. Conversely, if the measured value is higher than the desired value, the result is a frequency from the higher range. The further away the measured value is from the desired value, the lower or higher in frequency the audio signal becomes. The object is to adjust the relevant potentiometer to produce no tone—a result that occurs within a window around the desired value. Although the calibration process is intuitive, the program in Listing 1 visually prompts the technician about which potentiometer to adjust and in which direction to turn it. The technician quickly becomes accustomed to which speed, based on the audio frequency, he or she should adjust the potentiometer to and learns to slow the trimming as the measured value approaches the desired value. The program uses Turbo C++; Listing 1 shows only the relevant sections of code. (You can download Listing 1 from EDN's Web site: [www.edn-mag.com](http://www.edn-mag.com). At the registered-user area, go into the Software Center to download the file from DI-SIG, #2246.) (DI #2246)



## LISTING 1—AUDIO-CALIBRATION ROUTINE

```
#include <dos.h>
#define TOLERANCE 10

int tv,vt,in_range;
int channel;

//prototypes
void soundlo (unsigned int freq);
void soundhi (unsigned int freq);
int ADCreading (int channel);
int FetchSetPoint(void);

void main (void)
{
    ...
    ...
    ...
    tv=ADCreading(channel);
    //reading the A/D channel and conditioning it to the desired units
    vt=FetchSetPoint();
    //fetch the desired value
    in_range=0;
    //set flag for out of range
    if (tv>=vt)
    {
        //select frequency in upper range
        if ((tv-vt)>TOLERANCE)
        {
            //make a sound proportional in upper range
            soundhi(tv-vt);
        }
        else
        {
            //turn off sound
            nosound();
            in_range=1;
        }
    }
    else {
        //select frequency in lower range
        if ((vt-tv)>TOLERANCE)
        {
            //make a sound proportional in upper range
            soundlo(vt-tv);
        }
        else
        {
            //turn off sound
            nosound();
            in_range=1;
        }
    }
    ...
    ...
    ...
}

void soundhi (unsigned int freq)
{
    sound(1000+((freq)/2));
}

void soundlo (unsigned int freq)
{
    unsigned int temp;
    //checking and limiting minimum frequency to 100Hz
    if ((freq/2)>=400)
        temp=100;
    else
        temp=500-((freq)/2);
    sound(temp);
}
```

To Vote For This Design, Circle No. 359